Algorithms and Data Structures I

- Algorithm analysis techniques
- Basic data structures
- Searching
- Sorting
- Algorithm design techniques
- Graphs
- Weighted graphs
Algorithm analysis techniques

• Analysis methodologies
  - pseudo code, counting primitives, worst cases, ...

• Asymptotic notations
  - big/little-Oh, big/little-Omega, big-Theta
  - definitions and properties

• Mathematical tools
  - summations, log/exp identities, ...

• Justification techniques
  - (counter)example, contrapositive, induction, invariant,...

• Amortized analysis
Basic data structures

• Data structures
  - motivation, examples, applications
  - interface: ADT methods
  - implementation: design and analysis

• Basic data structures
  - stacks, queues, vectors, lists, sequences, ...
  - trees, tree traversals, binary trees, ...
  - priority queues, heaps, selection/insertion/heap-sort, ...
  - dictionaries, hash tables, hash collision, ...
Searching

• Searching in basic data structures
  − motivation, examples, applications

• Binary search trees
  − binary search tree properties
  − searching, insertion, removal
  − performance analysis

• AVL trees (balanced trees)
  − AVL tree properties
  − searching, insertion, removal
  − performance analysis
Sorting

• Merge-sort
  – divide-recur-conquer
• Set ADT
• Quick-sort
  – randomized quick-sort
• Sorting: a lower bound
  – comparison-based sorting
• Bucket-sort and radix sort
• Selection
Algorithm design techniques

• Greedy method: why greed works?
  − fractional knapsack
  − task scheduling
  − Dijkstra shortest-path tree

• Divide-and-conquer
  − recurrence equations
  − master theorem

• Dynamic programming: why not divide/conquer
  − 0-1 knapsack
  − Floyd-Warshall
Graphs

• Graph ADT
  – motivation, examples, applications
  – interface: Graph ADT
  – implementation: design and analysis
    • node, edge, adjacency

• Graph traversal
  – DFS, BFS

• Directed graph (digraph)
  – digraph traversal, transitive closure, DAG
Weighted graphs

• Motivation, examples, applications
• Single-source shortest paths
  – Dijkstra: why/when it works and how
  – Bellman-Ford
  – DAG
• All-pairs shortest paths
• Minimum spanning tree
  – Prim-Jarnik
  – Kruskal
  – Baruvka
What's next?

• Computer Science is about Algorithms
• More Algorithms courses from UVic CS
  - CSC320: Foundations of Computer Science
  - CSC326: Algorithms and Data Structures II
  - CSC425: Analysis of Algorithms
  - CSC426: Computational Geometry
  - CSC428: Computational Biology Algorithms
• Topics courses:
  • Graph Algorithms
  • Randomized Algorithms
What's more?

• Systems courses from UVic CS
  – CSC360: Operating Systems
  – CSC361: Computer Networks
    • changed from CSC450
  – CSC460: Embedded/Real-Time Systems
  – CSC461: Multimedia Systems
  – CSC462: Distributed Systems
  – CSC463: Wireless Mobile Networks
  – CSC465: Advanced Communication Networks
  – CSC466: Advanced Computer Networks

http://www.csc.uvic.ca/courses/undergrad.html